

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	4	jacob near gotwals.in.	US-PGPUB; USPAT	OR	ON	2007/02/08 17:42
S2	7	suresh near srinivas.in.	US-PGPUB; USPAT	OR	ON	2007/02/08 17:44
S3	15165	intel.as.	US-PGPUB; USPAT	OR	ON	2007/02/08 17:44
S4	11	S3 and (high\$1level and low\$1level).clm.	US-PGPUB; USPAT	OR	ON	2007/02/08 17:44
S5	3646	717/104-105,124-135.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/08 17:47
S6	185	S5 and (high\$1level and low\$1level)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/08 17:47
S7	124	S6 and model\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/08 17:48
S8	3	S7 and profil\$4 and (call\$4 adj graph)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/08 17:49
S9	118	S7 and (@pd<"20031216" or @ad<"20031216" or @prad<"20031216" or @rlad<"20031216")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/08 17:56
S10	6	("20020010913" "5995754" "6189141" "6374369" "6397379" "6622300").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/02/09 07:28
S11	13	("4860204" "5691895" "5890133" "6112126" "6182014" "6292810" "6411922" "6434435" "6442512" "6535795" "6560501" "6731998" "6778863").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/02/09 07:31

EAST Search History

S12	17359	high\$1level and low\$1level	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:11
S13	272	S12 and (profil\$4 near data)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:11
S14	227	S13 and model\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:11
S15	3	S14 and (hierarchical adj view)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:12
S16	5	S14 and (call\$4 adj graph)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:14
S17	212	S14 and (view or graph)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:14
S18	137	S17 and optimiz\$5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:15
S19	126	S18 and (@pd<"20031216" or @ad<"20031216" or @prad<"20031216" or @rlad<"20031216")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 14:34

EAST Search History

S21	6765	717/124-161.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:28
S22	524	S21 and (high\$1level and low\$level)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:28
S23	54	S22 and (profil\$4 near data)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 08:28
S24	16	("5151981" "5355487" "5797019" "5896538" "5949971" "5950009" "5960198" "6018759" "6026234" "6070009" "6226787" "6256775").PN. OR ("6519766").URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/02/09 08:37
S25	23	(architectur\$4 adj view) and (hierarch\$4 adj view)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 10:33
S26	7	("2002049573" "6311144" "5276877").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 10:33
S27	53	("5276877" "5446874" "5486995" "5522014" "5726914" "5771370" "5790789" "5801958" "5809282" "5881268" "5893074" "5953707" "5958009" "5984511" "5999734" "6003079" "6009256" "6038540").PN. OR ("6311144"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/02/09 10:35

EAST Search History

S28	97	("20050132336" "5968187" "20040064531" "6105112" "6173371" "6182201" "5729713" "5628014" "5542107" "5812935" "5917723" "5919267" "6277265" "6084471" "6289252" "6320600" "4809160" "5935255" "6246422" "5374966" "6094571" "6134430" "6498926" "5548718" "5737638" "5824933" "5912906" "5915288" "6055492" "6108536" "6115770" "6178499" "6289434" "6289443" "6820127" "20020165992" "20040185782" "20040215799" "20050086427" "20050210084" "20060085389" "5976990" "6111425" "6208645" "6587364" "6927930" "20050174670" "5745733" "20040098311" "20060184966").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 10:52
S30	2	"6668372".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 11:03
S31	2	"5950009".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 11:03
S32	79	717/133.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 14:34
S33	71	S32 and (@pd<"20031216" or @ad<"20031216" or @prad<"20031216" or @rlad<"20031216")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/09 14:34
S34	165	(expert adj system) with (advice or tip)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/10 09:19

EAST Search History

S35	4	S34 and "717".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/10 09:21
S36	1	S34 and (software adj performance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/10 09:22
S37	33	S34 and (software adj (application or system))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/10 09:22
S38	11	(model\$4 adj editor) and (model\$4 adj generator)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/10 09:29


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
[Search:](#) [The ACM Digital Library](#) [The Guide](#)

software performance analysis hierarchical model

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **software performance analysis hierarchical model**

Found 134,448 of 196,064

Sort results by relevance

[Save results to a Binder](#)

Display results expanded form

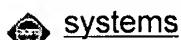
[Search Tips](#) [Open results in a new window](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

**1 Architecture based analysis of performance, reliability and security of software**

systems

Vibhu Saujanya Sharma, Kishor S. Trivedi

July 2005 **Proceedings of the 5th international workshop on Software and performance WOSP '05****Publisher:** ACM PressFull text available: [pdf\(411.28 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

With software systems becoming more complex, and handling diverse and critical applications, the need for their thorough evaluation has become ever more important at each phase of software development. With the prevalent use of component-based design, the software architecture as well as the behavior of the individual components of the system needs to be taken into account when evaluating it. In recent past a number of studies have focused on architecture based reliability estimation. But areas ...

**2 Performance evaluation of software architecture: Combining stochastic process****algebras and queueing networks for software architecture analysis**

Simonetta Balsamo, Marco Bernardo, Marta Simeoni

July 2002 **Proceedings of the 3rd international workshop on Software and performance WOSP '02****Publisher:** ACM PressFull text available: [pdf\(252.47 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We propose an integrated approach to the functional and performance analysis of Software Architectures (SAs) based on Stochastic Process Algebras (SPAs) and Queueing Networks (QNs), in order to combine their main advantages: formal techniques for the verification of functional properties of systems for SPAs, and efficient performance analysis for QNs. We first introduce *AEmilia*, a SPA based architectural description language for the compositional, graphical and hierarchical modeling of SAs, ...

**3 Middleware performance analysis: The metaPL approach to the performance****analysis of distributed software systems**

N. Mazzocca, M. Rak, U. Villano

July 2002 **Proceedings of the 3rd international workshop on Software and performance WOSP '02****Publisher:** ACM PressFull text available: [pdf\(141.69 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper presents an approach to the performance analysis of distributed software systems that relies on the use of the MetaPL notation system. MetaPL is an XML-based language that can describe programs written in different distributed programming languages and environments. The possibility to include timing information in the program

description promotes the use of software performance engineering techniques during software development. Moreover, MetaPL makes also possible to analyze and to t ...

Keywords: XML, distributed software systems, performance analysis, program description language, simulation

4 A cost-benefit decision model: analysis, comparison and selection of data management

 Stanley Y. W. Su, Jozo Dujmovic, D. S. Batory, S. B. Navathe, Richard Elnicki
September 1987 **ACM Transactions on Database Systems (TODS)**, Volume 12 Issue 3

Publisher: ACM Press

Full text available:  pdf(3.29 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes a general cost-benefit decision model that is applicable to the evaluation, comparison, and selection of alternative products with a multiplicity of features, such as complex computer systems. The application of this model is explained and illustrated using the selection of data management systems as an example. The model has the following features: (1) it is mathematically based on an extended continuous logic and a theory of complex criteria; (2) the decisi ...

5 Graph models for reachability analysis of concurrent programs

 Mauro Pezzè, Richard N. Taylor, Michal Young
April 1995 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 4 Issue 2

Publisher: ACM Press

Full text available:  pdf(3.00 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The problem of analyzing concurrent systems has been investigated by many researchers, and several solutions have been proposed. Among the proposed techniques, reachability analysis—systematic enumeration of reachable states in a finite-state model—is attractive because it is conceptually simple and relatively straightforward to automate and can be used in conjunction with model-checking procedures to check for application-specific as well as general properties. This article sho ...

Keywords: Ada tasking, process algebra, static analysis

6 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems

 Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker
December 1996 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on Model curriculum and guidelines for undergraduate degree programs in information systems IS '97**, Volume 28 Issue 1

Publisher: ACM Press

Full text available:  pdf(7.24 MB)

Additional Information: [full citation](#), [citations](#)

7 Performance analysis of software for an MIMD computer

 Connie U. Smith, David D. Loendorf
August 1982 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1982 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '82**, Volume 11 Issue 4

Publisher: ACM Press

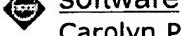
Full text available:  pdf(824.79 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper presents a technique for modeling and analyzing the performance of software

for an MIMD (Multiple Instruction Multiple Data) computer. The models can be used as an alternative to experimentation for the evaluation of various algorithms and different degrees of parallelism. They can also be used to study the tradeoffs involved in increasing the amount of parallel computation at the expense of increased overhead for synchronization and communication. The detection and alleviation of ...

8 A data modeling framework for the performance analysis of sequential and parallel software



Carolyn Pe Rosiene, Reda A. Ammar

March 1993 **Proceedings of the 1993 ACM conference on Computer science CSC '93**

Publisher: ACM Press

Full text available: [pdf\(690.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software Performance Engineering (SPE) is a modeling methodology that incorporates both functional and performance requirements into the development of high-performance, parallel, distributed, or real-time software. To aide SPE in achieving a framework suitable for modeling performance data, we present a data modeling framework that extends the object-oriented paradigm with modeling constructs necessary to represent semantics present in performance models. Two examples of semantic construct ...

9 H/S Embedded Systems: Performance analysis with confidence intervals for embedded software processes



Per Bjuréus, Axel Jantsch

September 2001 **Proceedings of the 14th international symposium on Systems synthesis ISSS '01**

Publisher: ACM Press

Full text available: [pdf\(269.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The choice of algorithms has a large impact on the performance of embedded real-time systems. Therefore, performance estimation of embedded software is vital in an early design phase. Consequently, high-level estimation techniques have been devised, but the accuracy of the estimations vary a lot depending on the algorithm and its context. We address this problem by proposing an estimation technique that both estimates the performance and computes the expected accuracy. The accuracy is used to pr ...

10 Performance modeling and analysis: Performance analysis of time-enhanced UML diagrams based on stochastic processes



Christoph Lindemann, Axel Thümmler, Alexander Klemm, Marco Lohmann, Oliver P. Waldhorst

July 2002 **Proceedings of the 3rd international workshop on Software and performance WOSP '02**

Publisher: ACM Press

Full text available: [pdf\(418.80 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

In this paper, we propose extensions to UML state diagrams and activity diagrams in order to allow the association of events with exponentially distributed and deterministic delays. We present an efficient algorithm for the state space generation out of these UML diagrams that allows a quantitative analysis by means of an underlying stochastic process. We identify a particular stochastic process, the generalized semi-Markov process (GSMP), as the appropriate vehicle on which quantitative analysi ...

Keywords: QoS performance modeling, model evaluation techniques, tools and techniques, transient and steady-state analysis of generalized semi-Markov processes

11 Performance by unified model analysis (PUMA)



Murray Woodside, Dorina C. Petriu, Dorin B. Petriu, Hui Shen, Toqueer Israr, Jose Merseguer

July 2005 **Proceedings of the 5th international workshop on Software and performance WOSP '05**

Publisher: ACM Press

Full text available: [pdf\(233.57 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Evaluation of non-functional properties of a design (such as performance, dependability, security, etc.) can be enabled by design annotations specific to the property to be evaluated. Performance properties, for instance, can be annotated on UML designs by using the "UML Profile for Schedulability, Performance and Time (SPT)". However the communication between the design description in UML and the tools used for non-functional properties evaluation requires support, particularly for performance ...

Keywords: UML, model building, performance models, scenarios, software performance engineering

12 Performance modeling from software components

 Xiuping Wu, Murray Woodside

January 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th international workshop on Software and performance WOSP '04**, Volume 29 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

When software products are assembled from pre-defined components, performance prediction should be based on the components also. This supports rapid model-building, using previously calibrated sub-models or "performance components", in sync with the construction of the product. The specification of a performance component must be tied closely to the software component specification, but it also includes performance related parameters (describing workload characteristics and demands), and it abst ...

Keywords: CBML, LQN, generative programming, layered queue model, performance prediction, software component, software performance, submodel

13 System level performance analysis: Formal performance analysis and simulation of UML/SysML models for ESL design

Alexander Viehl, Timo Schönwald, Oliver Bringmann, Wolfgang Rosenstiel

March 2006 **Proceedings of the conference on Design, automation and test in Europe: Proceedings DATE '06**

Publisher: European Design and Automation Association

Full text available: [pdf\(223.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

UML2 and SysML try to adopt techniques known from software development to systems engineering. However, the focus has been put on modeling aspects until now and quantitative performance analysis is not adequately taken into account in early design stages of the system. In this paper, we present our approach for formal and simulation based performance analysis of systems specified with UML2/SysML. The basis of our analysis approach is the detection of communication that synchronize the control fl ...

14 Verification: Model based estimation and verification of mobile device performance

 Gopal Raghavan, Ari Salomaki, Raimondas Lencevicius

September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04**

Publisher: ACM Press

Full text available: [pdf\(252.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Performance is an important quality attribute that needs to be planned and managed proactively. Abstract models of the system are not very useful if they do not produce reasonably accurate metrics. Detailed models are time consuming and expensive to build as well as to simulate. In order to strike a right balance, a framework is proposed in this paper that takes advantage of the flexibility of abstract modeling and intricacies of

detailed modeling. Performance is modeled and verified per use cas ...

Keywords: performance analysis, system level modeling, use case verification

15 Aspects of software design analysis: Concurrency and blocking



Connie Smith, J. C. Browne
May 1980 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1980 international symposium on Computer performance modelling, measurement and evaluation PERFORMANCE '80**, Volume 9 Issue 2

Publisher: ACM Press

Full text available: [pdf\(597.41 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper extends previous work on development of a methodology for the prediction of the performance of computer software systems from design level specifications and continuing through implementation. The effects of synchronized behavior, such as results from data reservation in multi-thread executions of data base systems, and competition for host system resources are incorporated. The previous methodology uses hierarchical graphs to represent the execution of software on some host comp ...

16 Shape-based retrieval and analysis of 3D models



Thomas Funkhouser, Michael Kazhdan
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: [pdf\(12.56 MB\)](#) Additional Information: [full citation](#), [abstract](#)

Large repositories of 3D data are rapidly becoming available in several fields, including mechanical CAD, molecular biology, and computer graphics. As the number of 3D models grows, there is an increasing need for computer algorithms to help people find the interesting ones and discover relationships between them. Unfortunately, traditional text-based search techniques are not always effective for 3D models, especially when queries are geometric in nature (e.g., find me objects that fit into thi ...

17 Software engineering and performance: a roadmap



Rob Pooley
May 2000 **Proceedings of the Conference on The Future of Software Engineering ICSE '00**

Publisher: ACM Press

Full text available: [pdf\(692.82 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: performance engineering, software engineering

18 Hierarchical model-based autonomic control of software systems



Marin Litoiu, Murray Woodside, Tao Zheng
May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2005 workshop on Design and evolution of autonomic application software DEAS '05**, Volume 30 Issue 4

Publisher: ACM Press

Full text available: [pdf\(393.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Various control algorithms are used in autonomic control to maintain Quality of Service (QoS) and Service Level Agreements (SLAs). Controllers are all based to some extent on models of the relationship between resources, QoS measures, and the workload imposed by the environment. This work discusses the range of algorithms with an emphasis on richer and more powerful models to describe non-linear performance relationships, and strong interactions among the system resources. A hierarchical framewo ...

Keywords: autonomic computing, performance models, self-management

19 Design and performance modeling of component interconnection patterns for distributed software architectures

 Hassan Gomaa, Daniel A. Menascé

September 2000 **Proceedings of the 2nd international workshop on Software and performance WOSP '00**

Publisher: ACM Press

Full text available:  pdf(276.11 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



Keywords: UML, XML, component interconnection patterns, performance model, queuing networks, software architecture

20 Facial modeling and animation

 Jörg Haber, Demetri Terzopoulos

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available:  pdf(18.15 MB) Additional Information: [full citation](#), [abstract](#)



In this course we present an overview of the concepts and current techniques in facial modeling and animation. We introduce this research area by its history and applications. As a necessary prerequisite for facial modeling, data acquisition is discussed in detail. We describe basic concepts of facial animation and present different approaches including parametric models, performance-, physics-, and learning-based methods. State-of-the-art techniques such as muscle-based facial animation, mass-s ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)